Design a mobile application to detect tomato plant diseases based on deep learning

Marwa Abdulla[1], Ali Marhoon[2]
[1]Department of Computer Science, College of Education, University of Basra, Basra, Iraq
[2]Department of Electrical Department, College of Engineering, University of Basra, Basra, Iraq

Article Info

Article history:

Received Apr 6, 2022
Revised Jul 7, 2022
Accepted Jul 27, 2022

Keywords:

Android studio
CNN models
Deep learning
Mobile application
Tomato disease

ABSTRACT

Plant diseases consider the most dominant matter for farmers' concerns because the operation of discovering and dealing with them requires accuracy, experience, and time. Therefore, this paper proposes an approach to classify seven varieties of tomato diseases using deep learning models. A dataset of 10448 images from PlantVillage and google utilize to train the deep learning (CNN models). The trained models proved their ability to classify with high accuracy, as the highest testing accuracy reached 95.71% for the proposed model for 50 epochs only. The resulted best model is published to a mobile application using the android studio platform, this application enables the farmer to classify plant diseases accurately and easily. The proposed model and mobile application could be extended to classify as many plant diseases as possible.

Corresponding Author:

Marwa Abdulla
Department of Computer Science, College of Education for Pure Sciences, University of Basra
Al-Ashar, Corniche Street, Basrah, Iraq
Email: cepsm510006@avicenna.uobasrah.edu.iq

1. INTRODUCTION

In particular, our country, Iraq, suffers from a noticeable decline in agricultural production after what was in the last century considered the primary source of the Iraqi economy [1], due to the reliance on traditional methods of agriculture and the lack of awareness of farmers about the diseases that effect on agricultural production. In general, farmers worldwide are in a race against time. The population increase in the world is growing at an accelerating pace, which requires more food in return for decreasing agricultural areas and climatic changes that the world is witnessing. all of these issues increase the challenges facing farms and crops. In the year 2050, the world population will rise to nearly 10 billion people, so the agricultural area must increase to cover the needs of the people and protect plants from pests, diseases, and weather factors that reduce plant production and quality [2].

All of the preceding factors have resulted in to use of IoT and deep learning, to provide accuracy and speed up the detection of diseases and pests counting fruits [3], [4]. Monitoring plant growth and predicting harvest time, and monitoring weather factors that affect plant growth [5], among the research that has been conducted in this field. Faithpraise and Chatwin [6] use k-mean clustering and correspondence filters to discover and distinguish the plant's location and the number of pests. It's worth noting that the correspondence filter can achieve rotational invariance of pests up to 360 degrees, demonstrating the algorithm's efficacy in detecting and recognizing lant pests. Sladojevic *et al*. [7] developed a model to distinguish plant leaves from their surroundings and identify 13 (thirteen) different plant diseases from healthy leaves, and Caffe used to perform the deep CNN training. Patil and Thorat [8] proposed work to develop a monitoring system, that will identify the chances of grape diseases in their early stages by using the

Hidden Markov Model and provides alerts via SMS to the farmer and the expert. The system includes temperature, relative humidity, moisture, leaf wetness sensor, and Zig-Bee for wireless data transmission. Padol and Yadav [9] using k-mean clustering and LSVM technique, to detect Powderly leaf and downy leaf disease for the grape plant. Wang *et al.* [10] used ResNet 50, VGG16, Inception v3, and the apple black rot images in the PlantVillage dataset, which botanists further annotated with four severity stages as ground truth, a series of deep convolutional neural networks trained to diagnose the severity of the disease. Rahnemoonfar and Sheppar [11] used a modified version of the Inception-ResNet architecture, there developed algorithm counts even if fruits occluded by foliage, branches, or some degree of overlap amongst fruits. Experimental results show 91% average test accuracy on authentic images and 93% on synthetic images. Wani [12] presented an idea on how to deploy WSN in the field and how the machine learning model fitted to predict pest/diseases using the Naive Bayes Kernel Algorithm. Devi *et al*. [13] used the k-mean, Random Forest Classification to discover and classify banana plant diseases. Khan *et al*. [14] used k-mean and CNN to investigate crop health or disease for 25 classes of plant diseases. Yadav *et al*. [15] utilized AlexNet, VGG16, VGG19, ImageNet, and YOLO-V3 networks to detect peach bacterial infections and reduce pesticides. Tiwari *et al.* [16] used DenseNet - CNN networks to classify diseases of 6 crops. Research by Natarajan *et al*. [17] faster R-CNN was used with ResNet to classify four types of tomato diseases. But the system confuses early blight with septoria leaf spot due to the lack of trained data 1090. Brahimi *et al*. [18] use the pre-trained models AlexNet, and GoogleNet to classify nine tomato diseases. GoogleNet achieved the highest classification accuracy of 99.185 and 97.711 with pre-trained and without pre-training sequentially. Durmus *et al*. [19] used AlexNet and squeezeNet models, and the dataset trained to classify ten tomato diseases with an AlexNet classification accuracy of 97.22, for squeezeNet, the accuracy was 94.3. Ashok *et al*. [20] use here image preprocessing to extract attributes with discrete wavelet transform (DWT) and gray-level co-occurrence matrix (GLCM). The images entered into the CNN image classification model, where the model achieved a classification accuracy of 98%.

Therefore, this paper used deep learning models to classify tomato diseases and create android mobile application using the classification model to be published. The research sections are as follows: The second section is the proposed method, the third section is the results, and the fourth section is the conclusion.

2. METHOD

In this paper, a mobile application based on deep learning was created to classify tomato leaf diseases when a picture of the leaf plant was taken with a mobile camera. Transfer learning techniques of VGG16 [21], VGG19 [22], and MobileNet_v2 [23], [24] models were used in training the data in addition to using a proposed model for CNN [25]. The model that achieves the best result in the test dataset is converted into a tensorflow lite model (TFLM) open source deep learning framework to run pre-trained models on Android or iOS. Figure 1 illustrate diagram for research method. The research methodology will be done through the following steps: assembling research tools, training and testing deep learning models, creating an application

2.1. Assembling research tools

In this research, we need the tools described below to implement it.

2.1.1. Dataset

CNNs need a data set for training, so the dataset of tomato disease of 11165 images for seven classes taken from PlantVillage for this type tomato late blight, early blight, leaf mold, yellow leaf curl virus**,** bacterial spot, healthy and mosaic virus. 129 images were collected from Google images for seven classes of tomato disease. Figure 2 shows images used in training where Figures 2(a) and 2(b) represent healthy, bactireal tomato disesases from PlantVillage and Figures 2(c) and 2(d) represent early, late blight tomato diseases from google image.

2.1.2. Android studio

The official integrated development environment (IDE) for the Android operating system from Google uses Java to develop applications [26]**.** This experiment was conducted on the latest version of the Bumblebee.

2.1.3. Experimentation

The experiment was carried out on a computer with an intel (R) core (TM) i5-4302Y CPU @1.60 GHz 1.60 GHz, 64-bit Windows 10 Pro, Python language, and open source machine learning platforms Tensorflow, Keras, And Google Colab.

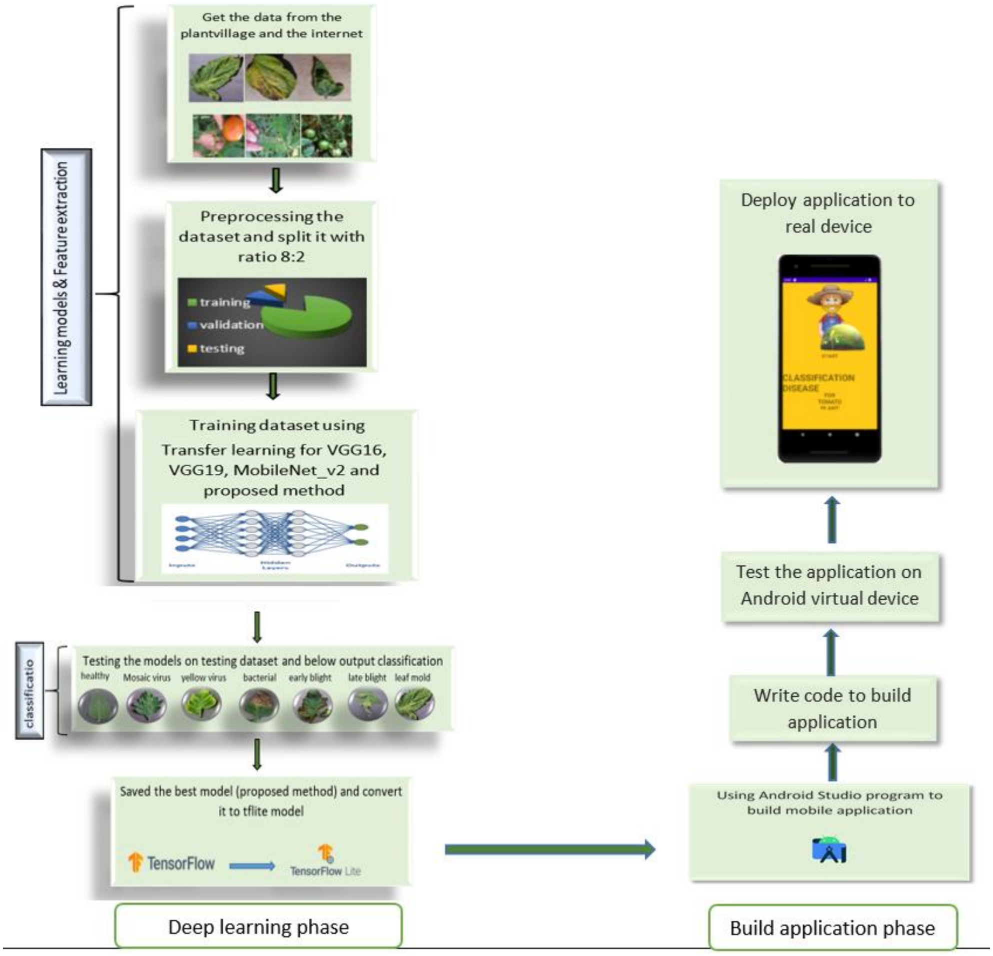

Figure 1. Diagram for research method

(a) (b) (c) (d)

Figure 2. Some image from training dataset (a) tomato healthy, (b) tomato bacrireal-spot, (c) tomato early plight, and (d) tomato late blight

2.2. Training and testing deep learning (CNN) models

As shown in Figure 3, which represents the work steps in this experiment. The work is in two parts, training the models on images and choosing the best model and converting it to TFLM [27] used to create an application using Android Studio to classify tomato diseases.

2.2.1. Data augmentation stage

The laboratory images of PlantVillage used in training have few features, containing only the image of the infected or healthy leaf of the plant. While field pictures are characterized by many features, for example, the presence of several leaves, the presence of a hand or foot taking the picture, the presence of fruits with leaves, etc. To increase the classification accuracy and make the model able to classify field images in addition to laboratory images. Collected 129 field images of tomato plant diseases from Google

Image. But the number of collected images is few to make the models learn from them during the training. So we did augmentation through the metrics rotation with 45 and 90 degrees, flipping horizontal and vertical, zooming with a value of 0.5, and brightness with a value between (0.5-1) to generate images. The generated images totaled 1642 and were added to the training dataset. Figure 3 shows augmented types on a single image from a dataset.

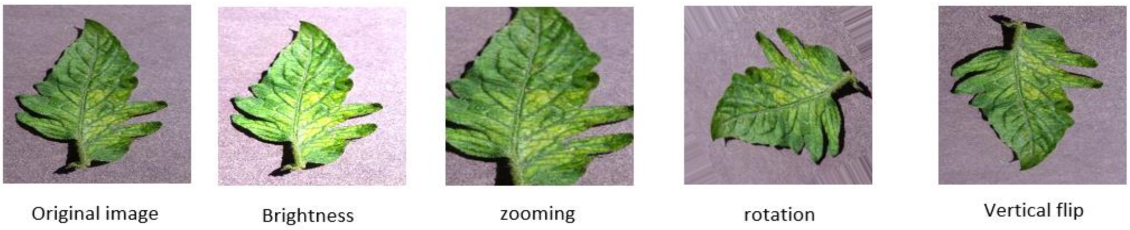

Figure 3. Methods for augmented image

2.2.2. Data pre-processing stage

Before applying deep learning algorithms, we need to preprocess the images by resizing the images from 256×256 to 224x224 for the pre-trained models and proposed method due the previously trained models were trained on the dimensions 224 × 224. Rascal the images by dividing them by 255 for compatibility with the initial values of the network. Split the dataset 80% as training data, 20% as validation, and the test dataset. Adding 1642 generated images from a set of field images collected from the internet to train models on a set of field images that contain more features than laboratory images, and 129 field images were added to the test dataset.

2.2.3. Models training stage

The proposed method used an image size of 224x224 in the input layer. As the ten hidden layers, the convolution process is applied to images using a kernel of weights 3×3 and a stride of 2. First, the activation function Relu collects the extracted attributes from the images and sends them to the next layer. After that, the data was transformed into a one-dimensional array by flattening in the fully connected layer and using the softmax activation function with seven classes as output. Figure 4 shows the structure of the proposed method.

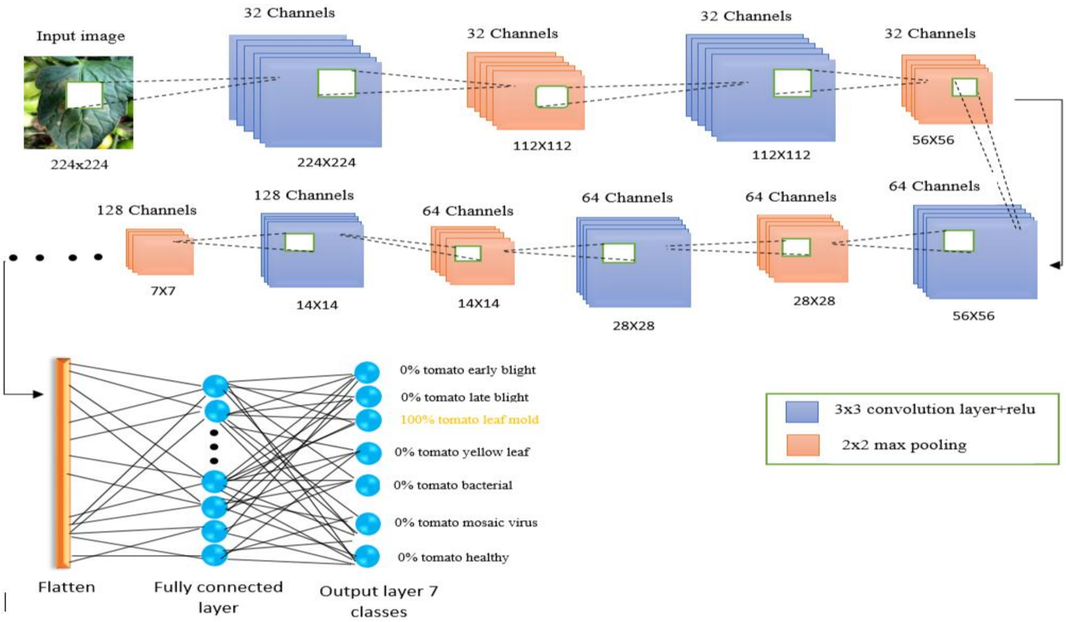

Figure 4. The structure of the CNN used in the proposed method

The data was also trained in a transfer learning method using models VGG16, VGG19, and MobileNet_v2, this is done by taking the weights of the pretrained models without the input layer and the upper layers, then we put the input layer of the training data set for tomato plants, and the upper layers are replaced by the dense layer, regular (normalization, dropout) using Relu activation function, last layer dense with 7 classes of tomato diseases and softmax activation function. The training parameters patch size=64, several epochs=50, and the Adam optimizer with categorical_crossentropy loss function for all the models. Table 1 showing the training and validation accuracy obtained for the models during training.

Table 1. Training and validation accuracy

Models	Training accuracy (%)	Validation accuracy (%)
VGG16	99.19	77.22
VGG19	99.41	94.9
MobileNet_v2	99.56	95.07
Proposed model	99.71	95.43

2.2.4. Models testing stage

After testing the trained models on the test dataset and getting the results shown in Table 2. It is clear that the best-calculated measures of network performance were obtained from the proposed model, where the highest value for the best measure of network performance f1-score for the proposed model, and the highest test accuracy with the lowest loss rate of 0.11458 obtained followed by VGG19, MobileNet_v2 and VGG16 with a loss rate of 0.1857, 0.1889, and 1.0018 respectively. After obtaining the best results from the proposed model, it was converted into TFLM to facilitate its storage and implementation and reduce its storage space for use in the next step in building the application. Figure 5 shows some of the test images for the proposed method.

Table 2. Performance of proposed method and transfer learning models

Model	Number of Images		Epochs	Test Accuracy (%)	Batch Size	Precision (%)	Recall (%)	F1-Score (%)
	Training	Validation and testing						
VGG16	10448	2488	50	77.69	64	85	77	78
VGG19				94.86		94	94	94
MobileNet_v2				94.94		93	94	93
proposed method				95.71		95	94	95

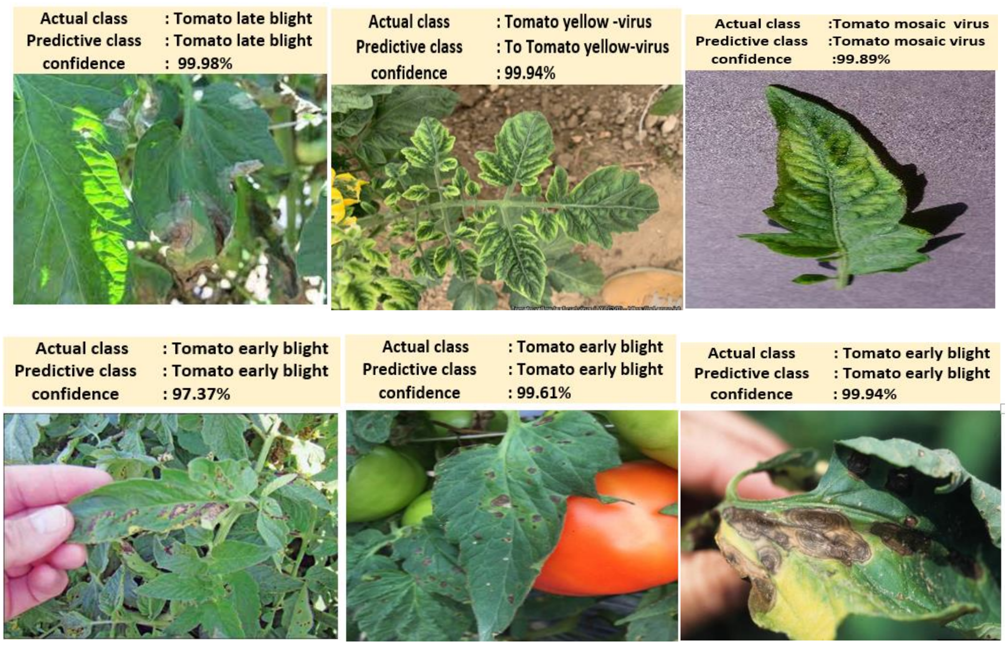

Figure 5. Some test results on the proposed model

2.3. Build application phase

To take advantage of the best model that is trained by the farmer to detect the type of disease that the tomato plant has. A mobile application was created, which published the best-trained model, android studio was used to create TFLM interpreter application with this screen:

2.3.1. Application start screen

The splash screen opens when the application starts. It contains a definition of the application's mission and a button on the classification screen.

2.3.2. Take picture and classification screen

The second screen is to take a picture of the plant leaf and display it on the screen with printing the name and efficiency of the classification compared to the list of classifications. Figure 6 shows the screen taking a picture of a plant leaf and classifying it.

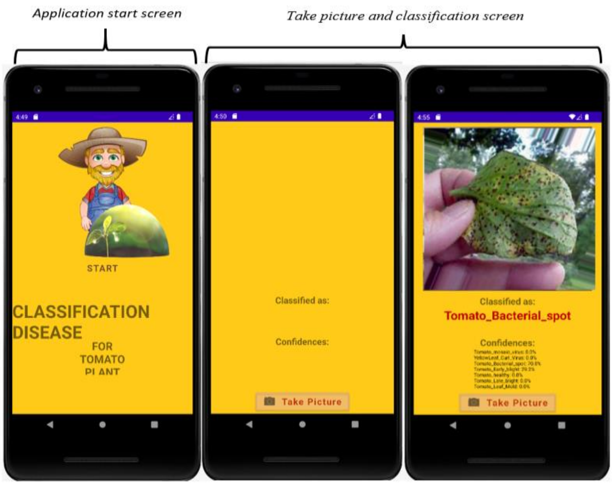

Figure 6. Screen for application

3. RESULTS

The application was tested to classify a group of tomato leaf diseases and the application worked efficiently in classification. Figure 7 shows how to classify a leaf after taking a picture with the mobile camera of a leaf displayed on the computer screen, although the picture is not clear when taken from the screen, it is classified correctly. Table 3 shows a comparison of work in this paper with some work in the same field.

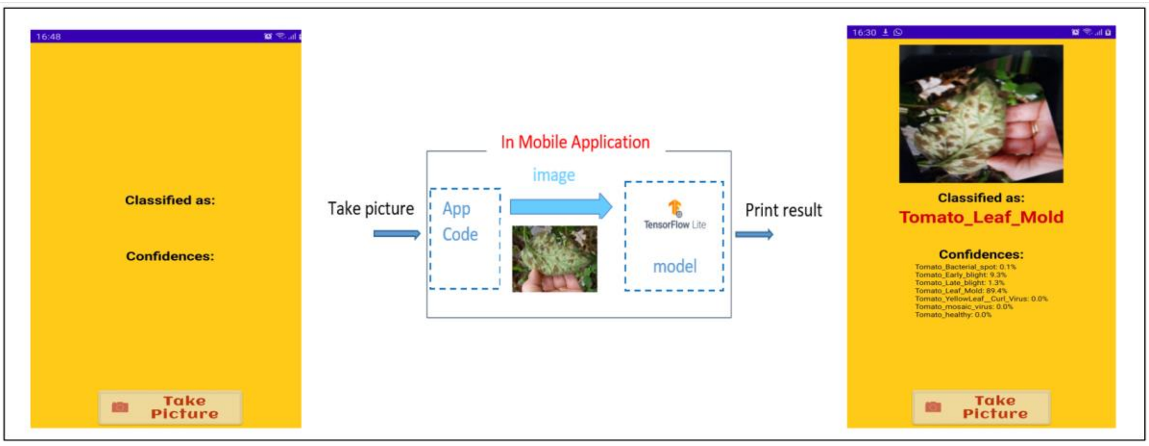

Figure 7. The result of the proposed work

Table 3. Comparing the experiment in this paper with related work

References	Image used	Classes	search work	Models used	Accuracy	Build app
[10] (Wang, *Sun, and J. Wang* 2017)	2086 images from PlantVillage	2	Assessment of the severity of apple plant disease	Proposed network with 8 convolution layers	79.3% for 12 runs	No
				VGG16	90.4 % for 12 runs	
[15] (Yadav, *Sengar, Singh, Singh, and Dutta* 2021)	2297 from PlantVillage and 28 images of bacteria	2	Determining the infection of the peach plant with bacteria	VGG16	93.75% for 70 epochs	No
				VGG19	94.61%	
				Proposed method	98.75%	
[16] (Tiwari, Joshi, and Dutta 2021)	25493 images from 4 dataset	27	Classification of plant diseases	MobileNet_v2	99.96% for 100epochs	No
				Dence Net 121	99.97%	
Experiment in this paper	11165 images from PlantVillage , 129 images from the internet,1642 generated image	7	Tomato plant diseases classification	VGG16	99.19% for 50 epochs	Yes
				VGG19	99.41%	
				MobileNet _v2	99.56%	
				Proposed method	99.71%	

4. CONCLUSION

In this study, deep learning was used in the agricultural field, especially in the classification of diseased tomato leaves. Pre trained models were used to train the data by transferring learning models VGG16, VGG19, and MobileNet_v2 the accuracy of the model test sequentially reached 77.69%, 94.86%, 94.94%. Then the data was trained on a proposed CNN network model to reach a higher accuracy of the test dataset, and when tested, its accuracy reached 95.71%, which led to its selection to convert it to the TFLM model to be used in Android Studio to create an application that classifies tomato leaf diseases running on the android system. And when testing the application in classifying plant leaves, its efficiency was good, so that when taking a picture of the leaf displayed on the computer screen, the application was able to classify it correctly most of the time, although the accuracy of photographing a leaf displayed on the screen is less than photographing a real leaf.

REFERENCES

[1] N. Al-Ansari, S. Ali Abed, and S. H. Ewaid, "Agriculture in Iraq," *In: Journal of Earth Sciences and Geotechnical Engineering*, vol. 11, no. 2, pp. 223–241, 2020, doi: 10.47260/jesge/1126.

[2] B. R. Yaseen, K. A. Al Asaady, A. A. Kazem, and M. T. Chaichan, "Environmental Impacts of Salt Tide in Shatt Al-Arab-Basra/Iraq," *IOSR Journal of Environmental Science, Toxicology and Food Technology (IOSR-JESTFT)*, vol. 10, no. 1, pp. 35–43, 2016, doi: 10.9790/2402-10123543.

[3] I. Mohanraj, K. Ashokumar, and J. Naren, "Field Monitoring and Automation Using IOT in Agriculture Domain," *Procedia Computer Science*, vol. 93, no. September, pp. 931–939, 2016, doi: 10.1016/j.procs.2016.07.275.

[4] B. T. Kitano, C. C. T. Mendes, A. R. Geus, H. C. Oliveira, and J. R. Souza, "Corn Plant Counting Using Deep Learning and UAV Images," *IEEE Geosci. Remote Sens. Lett*., pp. 1–5, 2019, doi: 10.1109/lgrs.2019.2930549.

[5] M. Abdulla and A. Marhoon, "Agriculture based on Internet of Things and Deep Learning," *Iraqi Journal for Electrical and Electronic Engineering*, vol. 18, no. 2, pp. 1-8, 2022, doi: 10.37917/ijeee.18.2.1.

[6] B. Faithpraise, C. Chatwin, F. O Faithpraise, P. Birch, R. C. D. Young, and J. Obu, "Automatic Plant Pest Detection And Recognition Using K-Means Clustering Algorithm And Correspondence Filters," *International Journal of Advanced Biotechnology and Research*, vol. 4, no. 2, pp. 1052-1062, 2013.

[7] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," *Computational Intelligence Neuroscience*, vol. 2016, 2016, doi: 10.1155/2016/3289801.

[8] S. S. Patil and S. A. Thorat, "Early detection of grapes diseases using machine learning and IoT," *2016 Second International Conference on Cognitive Computing and Information Processing (CCIP)*, 2016, pp. 1-5, doi: 10.1109/CCIP.2016.7802887.

[9] P. B. Padol and A. A. Yadav," SVM classifier based grape leaf disease detection," *2016 Conference on Advances in Signal Processing (CASP)*, 2016, pp. 175-179, doi: 10.1109/CASP.2016.7746160.

[10] G. Wang, Y. Sun, and J. Wang, "Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning," *Computational Intelligence Neuroscience*, vol. 2017, 2017, doi: 10.1155/2017/2917536.

[11] M. Rahnemoonfar and C. Sheppard, "Deep count: Fruit counting based on deep simulated learning," *Sensors (Switzerland)*, vol. 17, no. 4, pp. 1–12, 2017, doi: 10.3390/s17040905.

[12] H. Wani and N. Ashtankar, "An appropriate model predicting pest/diseases of crops using machine learning algorithms," *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2017, pp. 1-4, doi: 10.1109/ICACCS.2017.8014714.

[13] R. D. Devi, S. A. Nandhini, R. Hemalatha and S. Radha, "IoT Enabled Efficient Detection and Classification of Plant Diseases for Agricultural Applications," *2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, 2019, pp. 447-451, doi: 10.1109/WiSPNET45539.2019.9032727.

[14] F. A. Khan, A. A. Ibrahim, and A. M. Zeki, "Environmental monitoring and disease detection of plants in smart greenhouse using internet of things," *Journal of Physics Communications*, vol. 4, no. 5, 2020, doi: 10.1088/2399-6528/ab90c1.

[15] S. Yadav, N. Sengar, A. Singh, A. Singh, and M. K. Dutta, "Identification of disease using deep learning and evaluation of bacteriosis in peach leaf," *Ecological Informatics*, vol. 61, p. 101247, 2021, doi: 10.1016/j.ecoinf.2021.101247.

[16] V. Tiwari, R. C. Joshi, and M. K. Dutta, "Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images," *Ecological Informatics*, vol. 63, p. 101289, March 2021, doi: 10.1016/j.ecoinf.2021.101289.

[17] V. A. Natarajan, M. M. Babitha, and M. S. Kumar., "Detection of disease in tomato plant using Deep Learning Techniques," *International Journal of Modern Agriculture*, vol. 9, no. 4, pp. 525–540, 2020.

[18] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep Learning for Tomato Diseases: Classification and Symptoms Visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, 2017, doi: 10.1080/08839514.2017.1315516.

[19] H. Durmuş, E. O. Güneş and M. Kırcı, "Disease detection on the leaves of the tomato plants by using deep learning," *2017 6th International Conference on Agro-Geoinformatics*, 2017, pp. 1-5, doi: 10.1109/Agro-Geoinformatics.2017.8047016.

[20] S. Ashok, G. Kishore, V. Rajesh, S. Suchitra, S. G. G. Sophia and B. Pavithra, "Tomato Leaf Disease Detection Using Deep Learning Techniques," *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 2020, pp. 979-983, doi: 10.1109/ICCES48766.2020.9137986.

[21] J. Pardede, B. Sitohang, S. Akbar, and M. L. Khodra “Implementation of Transfer Learning Using VGG16 on Fruit Ripeness Detection,” I*nternational Journal of Intelligent Systems and Applications,* vol. 13, no. 2, pp. 52–61, April 2021, doi: 10.5815/ijisa.2021.02.04.

[22] M. Shaha and M. Pawar, "Transfer Learning for Image Classification," *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 656-660, doi: 10.1109/ICECA.2018.8474802.

[23] B. Chellapandi, M. Vijayalakshmi and S. Chopra, "Comparison of Pre-Trained Models Using Transfer Learning for Detecting Plant Disease," *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 2021, pp. 383-387, doi: 10.1109/ICCCIS51004.2021.9397098.

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. -C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

[25] V. Shah and N. Sajnani, “Multi-Class Image Classification using CNN and Tflite,” *International Journal of Research in Engineering, Science and Management*, vol. 3, no. 11, 2020, doi: 10.47607/ijresm.2020.375.

[26] N. Verma, S. Kansal, and H. Malvi, "Development of Native Mobile Application Using Android Studio for Cabs and Some Glimpse of Cross Platform Apps," *International Journal of Applied Engineering Research*, vol. 13, no. 16, pp. 12527–12530, 2018.

[27] H. Esmaeel and H. R. Esmaeel, "Apply Android Studio (SDK) tools," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 5, 2015.

BIOGRAPHIES OF AUTHORS

Marwa Abdulla holds a Bachelor's degree in Computer Science from the University of Basra, College of Education, Department of Computer Science. She is currently a master's student at the College of Education, Department of Computer science. Research interests in deep learning in agriculture. She can be contacted at email: cepsm510006@avicenna.uobasrah.edu.iq.

Ali Marhoon is PhD, Neuro-fuzzy Control Systems from Basrah University- Science College–Computer Science Dept. M.Sc. In Microprocessor-based control system from Basrah University–Engineering College–Electrical Dept, Iraq. B. Sc Electrical Engineering from Basrah University, Iraq. Currently Professor, Basrah University-Science College-Computer, Director of Computer Center–Basrah University, Iraq. Instructor in Cisco Networking Academy-Basrah-Iraq. research interests control systems, wireless sensor networks routing, wearable wsn, automatic inspection systems, mage processing. He can be contacted at email: ali.marhon@uobasrah.edu.iq.